Approaches to TFM Tools and Services Integration and Interoperability

The current Traffic Flow Management infrastructure has evolved over several decades, and recently a lot of progress has been made by CDM, AUA, S2K, and other organizations to build prototype TFM tools, applications, and concepts. Unfortunately, these efforts have resulted in a mix of hardware and software technologies that are essentially “stand-alone” capabilities. This presents an increasingly more difficult challenge to train, maintain, and effectively use these piecemeal technologies. Additionally, these stand-alone systems often need to reproduce certain basic services, such as flight information databases and map displays, that can lead to the duplication of effort and incompatible architectures as illustrated in Figure 1.

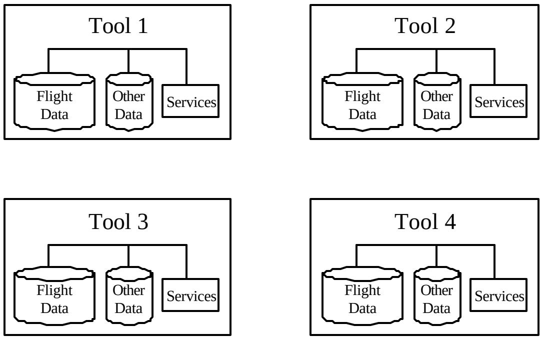

Figure 1: Current situation—stand-alone tools not integrated, and duplicating some services

Through recent efforts to improve training and operational procedures the FAA and the user community have clearly identified the need to integrate existing TFM tools and that new tools can no longer be built in a stand-alone environment. The question is how do we achieve integration, interoperability, and information sharing between tools that were/are being developed as stand-alone applications? One way is to create a master design and integrate all the individual applications into a single connected package (see Figure 2). This requires that custom interfaces be developed between each application that needs to communicate. This approach generally needs to be done all at once, and makes it difficult to integrate future applications into the overall tool set.

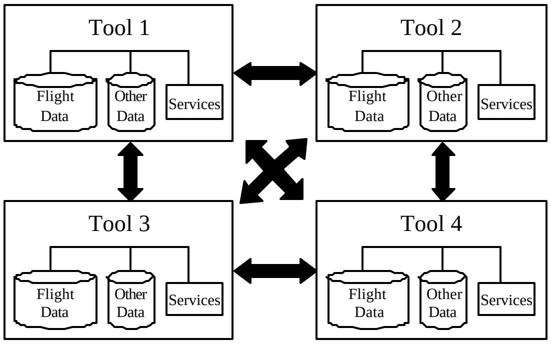

Figure 2: Integration achieved through developing custom interfaces between component applications

A second approach is adopt an architecture standard that requires applications to have a common APIs and information sharing standards (e.g., XML) that allow them to interact with other compliant applications and services (see Figure 3). This approach has several advantages:

- Each application need only to build a single API to be integrated into the common architecture
- Legacy applications can be wrapped (with middle ware) to provide the API
- Integration can be completed in phases allowing the most important applications and services to be integrated first
- Integration can be done in parallel removing many of dependencies between tools and different development organizations
- New applications and services built to the standard automatically become interoperable
- New/enhanced versions of existing applications and services can be deployed with less need to rework dependant applications (assuming the new version continues to adheres to the interface standard).

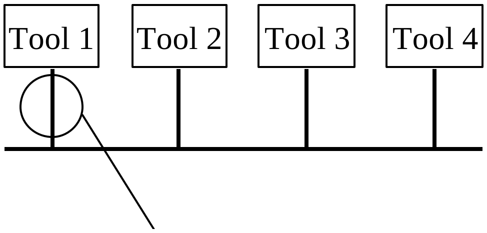

Figure 3: Integration achieved through developing API standards that allow applications to interface other compliant applications

Another big advantage is that this approach also allows the eventual pulling out of common services from individual applications for use by other applications. This has the advantage of reducing duplication of function and the need for separate maintenance, which can lead to synchronization problems (e.g., different applications using different sources of the same NAS configuration data). Figure 4 illustrates the potential result of such an approach to integrating TFM components.

Agreement on the particular standards by the FAA and the various development organizations would be made easier if the architecture adopted is based on open-standards (e.g., XML, ODBC/JDBC, SQL, CORBA, SOAP, J2E, EJB, etc.). These standards are widely used in commercial development, and would allow FAA systems to benefit from the technological advances being made in the commercial sector.

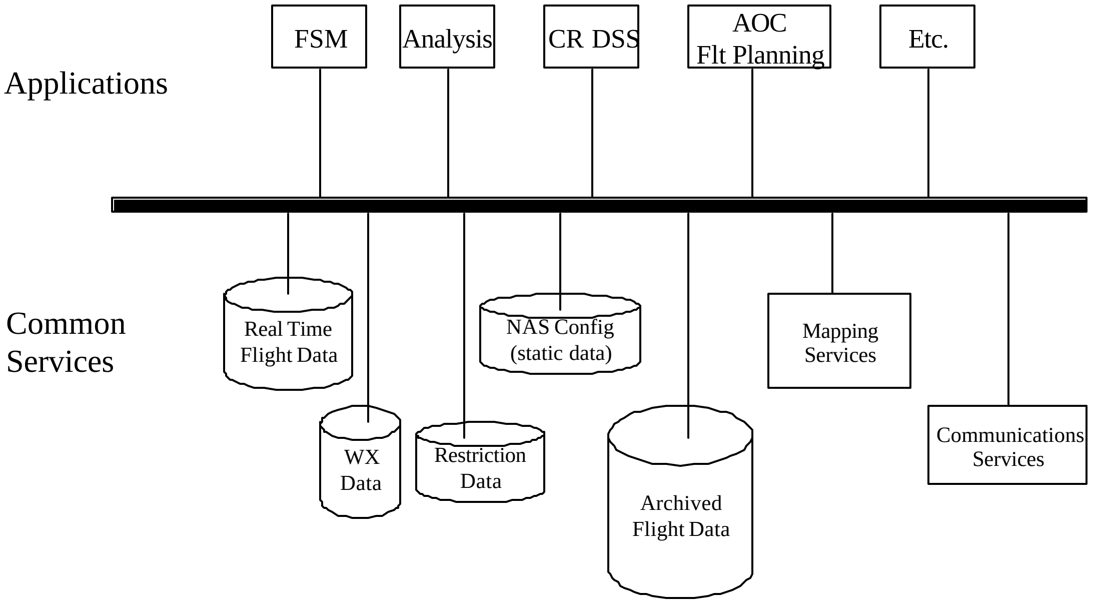

Figure 4: Potential evolution of open-standards based architecture to future integrated TFM environment where applications share common services